Word2Vec using Skip-Gram Model

Danish Shaikh
M.Tech (ECE)
shaikhm@iisc.ac.in

Abstract

Word2Vec is used for word embeddings. These models are usually two-layer neural networks that are trained to embed vectors for every unique words in the dataset. I have used Skip-Gram Model to implement Word2Vec model with window size = 2. Word2Vec takes a large corpus of text as an input and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors that are in proximity, with respect to context, to one another are closely positioned in the vector space.

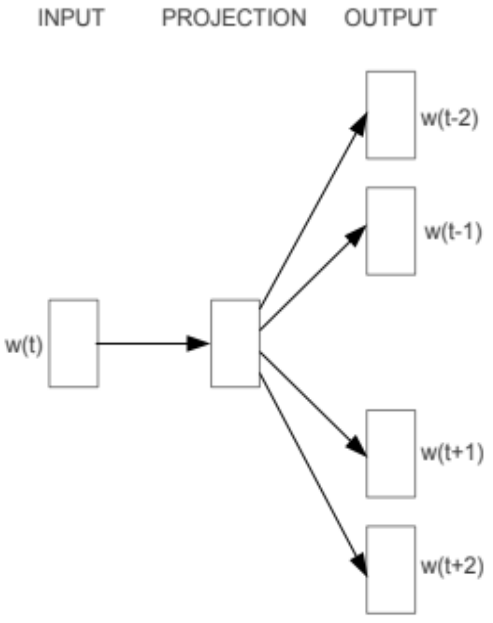
1 Introduction

Word2Vec is used for many applications such as analogy tasks, creating rhymes, radiology, bioinformatics,etc. Word2vec can utilize either of two model architectures to produce a distributed representation of words: Continuous Bag-Of-Words (CBOW) or Continuous Skip-Gram. In the Continuous Bag-Of-Words architecture, the model predicts the current word from a window of the surrounding context words. The order of context words does not influence prediction in Continuous Bag-Of-Words. In the continuous skip-gram architecture, the model uses the current word to predict the surrounding context words (N-words before the index word and N-words after it). The skip-gram architecture weighs nearby context words more heavily than more distant context words.

Word2Vec highly depends on the hyperparameters such as batch-size, embedding-size and negative samples. I have tweaked some combinations of them and obtain the highly Spearman's Correlation[4]. The achieved value in word2vec paper[1] for Spearman's Correlation is 0.28. **The value for Spearman's Correlation that I obtain is around 0.16 for Reuters Dataset**

2 Skip-Gram Model

The Skip-gram model architecture usually tries to achieve the reverse of what the CBOW model does. It tries to predict the source context words (surrounding words) given a target word (the center word). Considering our simple sentence from earlier, ***the quick brown fox jumps over the lazy dog***. Now considering that the skip-gram models aim is to predict the context from the target word, the model typically inverts the contexts and targets, and tries to predict each context word from its target word. Hence the task becomes to predict the context [quick, fox] given target word brown or [the, brown] given target word quick and so on. Thus the model tries to predict the context-window words based on the target-word. model.png



Skip-gram

3 Assignment Tasks

The Assignment consists of two tasks:

3.1 Task 1:

In Task 1, the neural network model is trained for word embeddings considering Reuters Dataset with the help of Tensorflow. The hyperparameters that I tweaked to get highest Spearman's Correlation were batch size, embedding size, negative samples, window-size.

3.2 Task 2:

Task 2 consists of analogy task where we have to get the relationship from the first 2 words and predict the forth word from the third word and that relation.
For this we have used "questions-word.txt" as mentioned in the guidelines of the Assignment.

3.3 Bonus Task:

This task is to check biases in the learnt word-embeddings.
For this, I have obtained the top 10 most nearest neighbours of the words 'man' and 'woman'.

4 Implementation

I have used ML libraries such as tensorflow, nltk, keras, etc. for dataset, pre-processing the data and embedding the words.

- Task 1:
    1. Download the Reuters dataset and import using nltk in python.
    2. Extract words from the sentences and create pairs of words using window of size 2.
    3. Create vocabulary containing unique words from the corpus.
    4. Make pairs of words with window size = 2.
    5. Pass the first array of the pair of words to the neural networks as inputs and second array as labels.
    6. Train the model using Neural Networks of 2 layers with the help of tensorflow.
    7. Minimize the loss function (NCE-loss) while applying stochastic gradient descent algorithm.
    8. Write the embeddings of the unique words in the corpus in a file.
    9. Validate for hyperparameters on SimLex-999 word similarity task.

- Task 2:
    1. Follow similar steps from Task 1 for the best model.
    2. For analogy task, use the below formula:
       Consider the analogy
       Athens : Greece :: Baghdad : ?

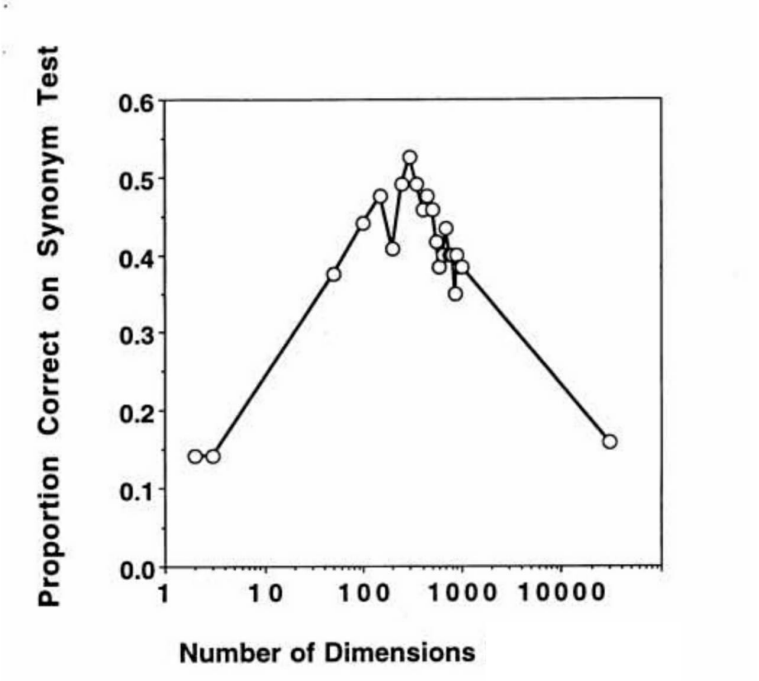
$$\underset{x}{\operatorname{argmax}}\, f(x) \tag{1}$$

where

$$f(x) = \frac{sim(x, Baghdad) * sim(x, Greece)}{sim(x, Athens)} \tag{2}$$

5 Problem of Underfitting and Overfitting

Depending on the dataset we should select hyperparamets to avoid underfitting and overfitting.
For this model, I have observed that if we increase embedding size beyond 300, the performance starts to decrease in terms of Spearman's Correlation.
To support this claim, I have attached the graph of Correlation vs. Dimension size (source : Internet).



Batch-Size	Negative-Samples	Embedding-Size	Spearman's Correlation
32	2	128	0.0143
32	2	64	0.0675
32	32	128	0.0974
32	6	64	0.1281
64	32	64	**0.1598**
128	16	200	0.0874

Table 1: Tuning of hyperparamters for optimal model.

6 Analysis

6.1 Task 1:

I have validated the model on SimLex-999 dataset, keeping window-size = 2 with epochs = 100.
The results are tabulated above for different hyperparameters:

6.2 Task 2:

For this task, I have used "questions-words.txt".
The task is to predict the forth word uby obtaining the relationship between first two words.
Consider the analogy:
London:England :: Athens: ?
The top 10 words come out to be :
['English', **'England'**, 'Portugal', 'Norwegian', 'Cuba', 'Albania', 'Mexican', 'Belarus', 'Spain']
As we can see that 2nd word comes out to be the desired result. Hence our model can be used for analogy task.

6.3 Bonus Task:

For this task, we have to check for biases in the learnt word-embeddings.
Consider top 10 nearest neighbours of the words **'king'** and **'queen'**
For **'king'** :
['stepbrother', 'father', 'he', 'sisters', 'groom', 'grandson', 'uncle', 'bride', 'brother']
For **'queen'**:
['stepdaughter', 'bride', 'stepbrother', 'sisters', 'groom', 'granddaughter', 'mother', 'aunt', 'brother']

As we can see, the model is kind of biased with respect to gender. Many papers have been published to eradicate this biasedness (*See References*)

7 Conclusion

- This Word2Vec model using Skip-Gram Model learns word-embedding from the corpus and represent each unique word in the corpus in the vector space to help in distributed learning.
- This model also helps us to identify the analogy between the words and can perform analogy task.
- This model is kind of biased with respect to gender.

8 Github

All the files are uploaded in the Github repositories. The link is provided here.

9 References

Word2Vec paper as mentioned in the guidelines: word2vec
J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: Theres software used across the country to predict future criminals. and its biased against blacks., 2016.
Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, Adam Kalai: Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings